The Design and Implementation of the VMOC Prototype

Robert Bane
Global Science and Technology, Inc.
6411 Ivy Lane, Suite 610
Greenbelt MD 20770
USA
bane@gst.gsti.com

Jeffrey A. Fox
Pacific Northwest National Laboratory*
901 D Street SW, Suite 900
Washington, DC 20024
USA
ja_fox@pnl.gov

In response to the pressures to reduce the cost of mission operations, the Data Systems Technology Division (Code 520) of the Mission Operations and Data Systems Directorate at NASA Goddard Space Flight Center has been building a new operations concept called the Virtual Missions Operations Center (VMOC). The VMOC concept calls for a system that provides flexible distributed support for mission operations, in which routine tasks are automated, and dynamically configured teams act as on-demand support personnel to aid or take control of the system as needed. This paper focuses on the tools being used to develop the VMOC; in particular the G2/IMT expert system and Lotus Notes. It describes why the tools were selected, how those tools are used in designing the VMOC, lessons learned from working with those tools, and a brief discussion on the future role of those tools as the VMOC matures.

INTRODUCTION

Today, operations make up a considerable proportion of the life cycle cost of spacecraft. Space operations are expensive, in part, because current mission operations centers require dedicated facilities and continuous monitoring by human operators. To address these problems, a new paradigm for spacecraft operations, called the Virtual Missions Operations Center (VMOC), has been developed under the sponsorship of NASA Goddard Space Flight Center in Greenbelt, MD, USA.

In the VMOC, spacecraft management will be conducted by dynamically configured teams who act as on-demand supervisors at any time in any place. The supervisory tasks will take advantage of increased automation that would allow for the implementation of a proactive management-by-exception paradigm [1]. During routine operations, standard process monitoring and management tasks will be performed autonomously using advanced automation, expert systems, and software agents. However, when a potential critical fault or emergency is detected, workgroup computing tools allow the dynamic creation of a response team by identifying and adding the most appropriate personnel and resources from remote locations. A graphical representation of the VMOC components is shown in figure 1.

*Pacific Northwest National Laboratory is operated for the U.S. Department of Energy by the Battelle Memorial Institute under Contract DE-AC06-76RLO 1830.

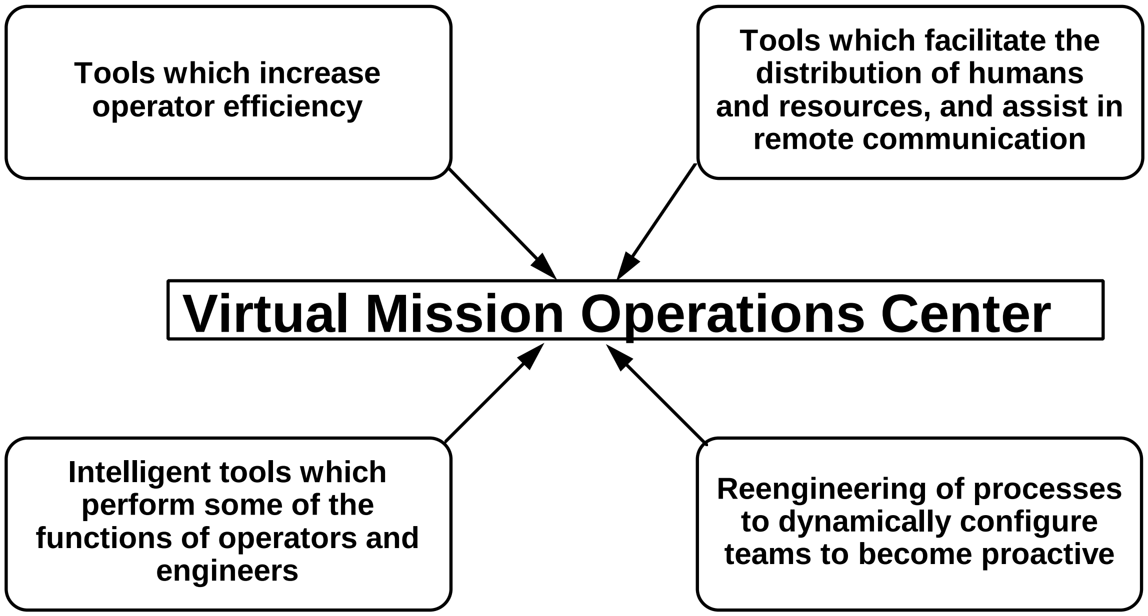

Figure 1. VMOC components

To accomplish the goals of the VMOC and build the necessary tools, the VMOC system utilizes both commercial off-the-shelf and custom developed software. The key applications being used to implement the VMOC are:

- G2™ and Intelligent Mission Toolkit™ (IMT) -- an expert system shell and extensions for automated spacecraft monitoring and command and control.

- Lotus Notes™ -- groupware for workflow automation, automated form filling and management, on-line documentation, threaded communications, and built-in distributed paging and phone-based interfaces.

The remainder of this paper focuses on the VMOC development team's experiences with these tools. A brief overview of the tools' capabilities is presented. This is followed by a discussion on how the tools are used within the VMOC, the lessons learned from using those tools, and the future role of those tools in the development effort.

Designing with G2 and IMT

Automation of the routine spacecraft monitoring and maintenance is an important part of the VMOC concept. Expert systems are a mature technology well suited to the implementation of monitoring and control systems. Many vendors sell expert system wrighting tools (called "shells"). For the VMOV propotype, we chose a COTS spacecraft monitoring and control system based on an expert system shell; this allowed us to buy a basic automation framework and concentrate our efforts on adding distributed capabilities

G2

[Add a couple of sentences on how expert system (in general) is part of VMOC]

G2 is a "kitchen-sink" philosophy expert system shell from Gensym Corporation. It allows system development using procedures, rules, objects, fuzzy logic, and graphics; all its features are well integrated. All development is done within G2 using its proprietary tools (editor/GUI/debugger). G2 runs under Unix, VMS, Windows NT, or MacOS. G2 developed systems are portable across platforms with no source changes.

IMT

Intelligent Mission Toolkit (IMT) is a spacecraft commanding and monitoring system built on top of G2 by Storm Integration. IMT has a graphical object-oriented commanding system. Each pass-plan step is an object; steps can represent simple operations (send one spacecraft command) or complex ones (run an expert system rulebase). Pass plans are built by taking objects and connecting them together into diagrams that look and function like flow charts. A completed pass plan has an object-and-link structure that can be easily traversed and manipulated by other G2 code, so plan analysis is straightforward. IMT supports the Loral 500 front end processor for command transmission and telemetry. Mappings between commands and bit-string encodings are stored in a Sybase database.

[Add graphic of O-O] pass plan?]

G2/IMT was chosen as the platform for the VMOC prototype primarily because of a G2 capability called TeleWindows, which allows multiple users to see and interact with a single running inference engine. This capability is critical to the VMOC concept of remote expert collaboration. *[Explain more here.]*

VMOC Extensions to G2/IMT

In implementing the VMOC prototype, we extended the standard IMT shell by added more automation in pass-plan execution as well as gateways to allow the system to react to events outside of spacecraft telemetry.

Specifically we added:

- Pass-plan queuing -- structures where pass-plans can be queued to wait for future events like acquisition of spacecraft signal or availability of a resource. Storm Integration has since added a similar feature to IMT.

- Pass-plan analysis -- a facility for stepping through pass-plans, noting arbitrary features, and processing them the plans accordingly.

- Email gateway -- a facility that allows IMT to send and receive email; appropriately formatted incoming mail can trigger events within IMT.

- Data structures -- prototype representations of experts, areas of expertise, and mappings between them and detectable spacecraft problems.

Sample Scenario

A pass-plan is submitted to IMT, possibly via email, and queued for future execution. VMOC analyzes the submitted plan and notes any special resources it may require (e.g., expert human monitoring). VMOC determines the availability of those resources by querying databases or sending email to required personnel, who can agree to being assigned by return email. When all needed resources are allocated, VMOC automatically releases the analyzed plan for execution at its requested time. If a problem is detected during execution, VMOC classifies it as to type and severity, ~~chooses experts to handle it, and notifies them of it using email or pager, depending on the problem's severity~~ and then emails a notification of the event to the Lotus Notes subsystem so that it can contact the appropriate experts. The experts can then communicate with VMOC and each other via TeleWindows to diagnose the fault.

Lessons Learned

When the VMOC project started back in 1994, G2/IMT was a good choice as a platform, because of its support for multiple display clients from one inference engine and its spacecraft-specific focus. Its non-standard GUI was seen as a small drawback compared to its other features, and its full-featured development environment was seen as a plus for prototyping despite its complexity.

Since then, other alternatives have become available, both within NASA and commercially. Two of the most interesting ones are ALTAIR and GenSAA/Genie. ALTAIR is a spacecraft commanding and monitoring system from ALTAIR Aerospace. It can do VMOC-style networking from its RTworks base, and its state-based modeling looks like an intelligent tool that can be maintained by engineers and operators instead of computer scientists. This is an important consideration when moving from prototype to deployment. ALTAIR has attracted a lot of attention at NASA Goddard recently, primarily because of the IMACCS 90-day, all-COTS ground system project (see **http://joy.gsfc.nasa.gov/SpOps96/gse/sp96228.txt**).

Generic Spacecraft Analyst Assistant (GenSAA) is under development by Code 520 of NASA at Goddard. It started out as a tool to make it easier for flight operations team (FOT) personnel to generate intelligent telemetry displays, and handles that very well. GenSAA essentially connects the CLIPS rule engine to spacecraft telemetry via TPOCC, and lets users drive GUI widgets from the rule engine. Since the start of VMOC, GenSAA has added Generic Inferential Executor (Genie), a system that lets the rule engine do commanding and monitoring. GenSAA/Genie originally had no support for VMOC-style networking, but has recently added socket-based access to the rule engine, so the infrastructure is there (see http://groucho.gsfc.nasa.gov/Code_520/Code_522/Projects/GenSAA/).

Recently the VMOC developers have done a study comparing these three systems as potential future platforms for VMOC; for details **see <http:wherever-my-TEAS-stuff-is>**.

[Short piece to tie things up...conclusion/future]

LOTUS NOTES

In analyzing current operation centers, it became clear that to maintain the same level of performance as conventional operations centers while drastically reducing the number of staff, the VMOC needs more than an export system. Additional software would be needed to facilitate the routine non-real-time MOC activities (e.g., pass planning as well as fault management activities, like anomaly resolution). What makes designing a system to support these features such a challenge is the fact that the system will support a team that is now distributed both in place, as well as time. Additionally, the team members will no longer have frequent interaction with the system since most of the team will only be utilized as on-call personnel.

These design requirements lead to the decision to include a groupware system as the infrastructure or glue for the VMOC operation concept. Groupware refers to the computer facilities for groups, usually small, or organizations that enables or supports them in electronically achieving their shared goals. Typically, groupware supports one or more of the following components:

Communication -- the ability to transmit and receive information
- » Tools: Electronic discussion and Video conferencing

Collaboration -- the sharing of information to reach a common goal
- » Tools: Workflow management and Scheduling

Coordination -- the support and management of group work and tools
- » Tools: Message & document databases and meeting facilitation

Because the design team wanted to utilize COTS products wherever possible, Lotus Notes was chosen as the groupware tool. Though not currently used in the command and control community, it is the de facto standard for groupware in the business community. Lotus Notes is a client/server groupware package with the core capabilities to build custom applications, send and receive message (email), access multiple mixed document databases, and provide workflow automation. Additionally, there are a large number of third-party tools that increase Notes' functionality, including Internet (World Wide Web) access and publishing, video conferencing, paging, and telephony access.

Currently, the Notes development effort is focusing on the fault detection, isolation, and recovery (FDIR) aspect of mission operations. This aspect of operations was chosen because it supports the most immediate needs of current missions need to eliminate operators to stay active. The basic Notes FDIR architecture is shown in Figure X.

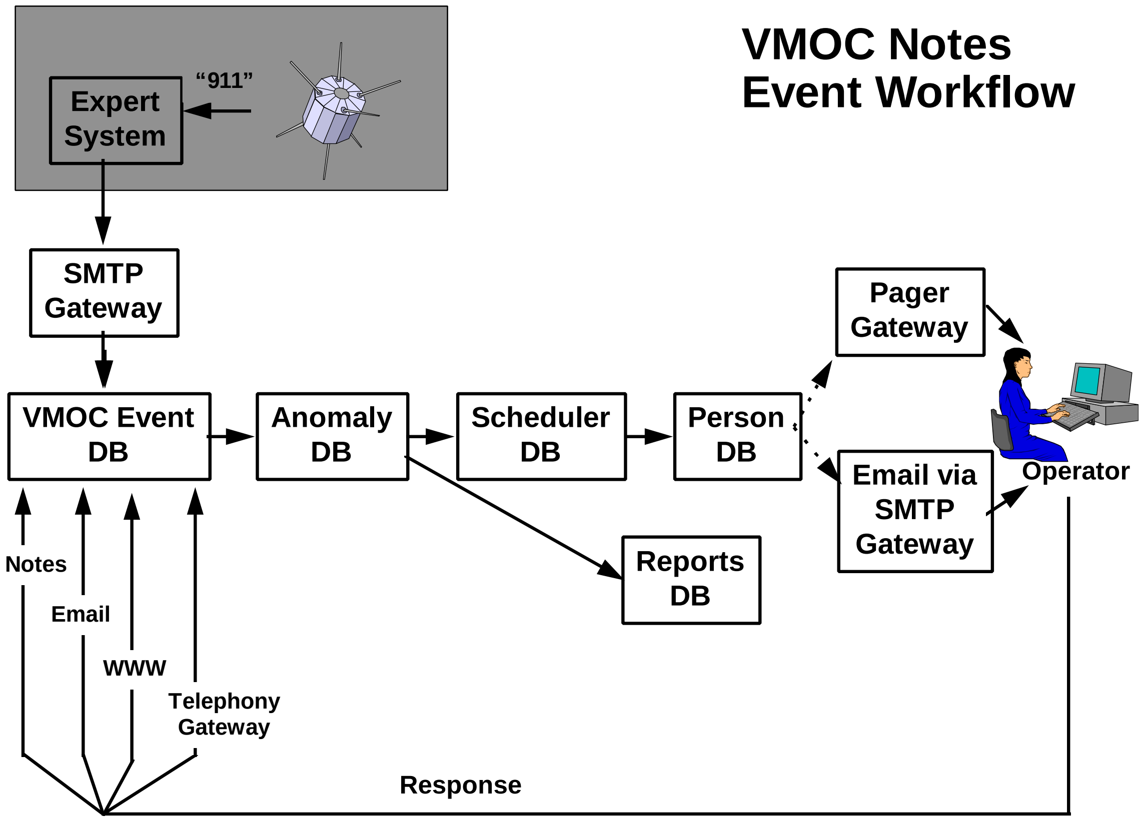

Figure X: Basic Notes Architecture for supporting FDIR

In this architecture, Notes relies on the expert system to do all the real-time data processing and fault detection. Once the expert system identifies an anomaly it will email Notes with a simple email message (through an SMTP gateway) that contains the event information, such as the type of fault, time identified, and pass number. Using the email delivery mechanism, the Notes part of VMOC can communicate with any expert system that can generate an email message (e.g., via SendMail). Once Notes receives the email message, an agent parses the text message and stores that data under a unique ID in the *Event* database.

The agent then uses that event (fault) ID code from the email message to look up the appropriate response notifications based on the data in the *Anomaly* database. That anomaly database contains a unique record for each known type of anomaly. For each type, there is a record containing a text description of the problem, the associated security level, and notification information. The notification data lists the types of people to contact (e.g. operator of engineer), the priority of contacting them (e.g. emergency or standard), and the alert message to be sent.

Once the agent reads in the type of people to notify, it will open the *Scheduler* database to get the names of the team members who are on-call at that time. After collecting the names, the agent will search through the *Person* database to collect the appropriate contact media (e.g. pager) and contact numbers (e.g. pager service and PIN numbers).   It will then contact the team members via the appropriate gateways (either pager or SMTP for email).

Another agent will then begin monitoring to see if the on-call team members login to the Notes database to acknowledge their notifications. If there is no acknowledgment within a specified period of time (as defined in the *Anomaly* database), there will be a roll-over notification sent to another person. This roll-over will continue until somebody replies to their alert notification.

Once he/she receives an alert notification, the team member can access the *Event* database through a local Notes client, a web browser (via Domino with SSL) or a telephone using a third-party product called MailSpeakIt.

MailSpeakIt is a tool that provides remote and mobile users access to their Lotus databases via a touch tone telephone. MailSpeakIt utilizes electronic mail, speech, and telephony server technology to enable users to read, create and send Lotus Notes and electronic messages with both text and/or recorded voice mail message and attachments. It also allows user full command and control of their databases.

Once the FDIR component is complete, the role of Notes will be greatly expanded. Notes will be used in all aspects of mission operations, from providing a web-based front-end for principal investigators to requesting data collection for post-pass administrative functions. In fact, there are plans to try to use Notes throughout the life cycle of a spacecraft by also facilitating design activities via databases for design documentation and operators' manuals. This concept should help reduce the cost of a mission by increasing the continuity of the program.

Lessons Learned

To date we have found Lotus Notes to have tremendous potential for supporting highly automated space operations. Notes' built-in scripting language and agents allows for easy creation of workflows. Also, the ease of integration with alternative communications devices (e.g., pagers) makes fault notification fairly straight forward to implement. However, there have been, and still are, significant obstacles to overcome. In particular, Notes has no built-in capability to support agent triggering actions at a resolution of less than one half hour. This initially prevented us from easily building the hierarchic roll-over paging scheme. Lotus claims this feature will be changed in future releases on Notes. Until then, we are experimenting with a more complex mechanism of utilizing multiple agents that work together at fixed time intervals. If this does not work, API-level programming will be required. However, this would somewhat reduce the advantage of using a COTS product.

Also, we have found, like many others who are just beginning to use Notes for complex tasks, it is worthwhile to use a certified Notes consultant to assist in designing the overall architecture, selecting and interfacing with the right third-party tools, and assisting in the development of critical or unusual functionality. Lotus Notes is so unlike traditional database design that it takes time for in-house developers to become trained in Notes and to develop a sufficient level of proficiency. It is during this ramp-up time when an experienced Notes developer is most useful.

The final lesson learned is that once people see the potential for using Lotus Notes, there is a desire to use it for a variety of non-project related tasks (e.g., an automated help desk database). Unless supplemental tasks are appropriately prioritized and staffed, they can take away from the main goal of the project. On the other hand, it is never desirable to extinguish excited over a new tool.

CONCLUSION

The VMOC prototyping effort is still on-going today. Once the core infrastructure elements are completed, the goal is to find several missions with which to team. The goal of teaming with a missions is to test the VMOC software in an operational environment and to hopefully validate many of its operational concepts. Lessons learned will be used to reevaluate tool selection, optional concepts, and software and user interface designs. From there, it is envisioned that the VMOC software would be iteratively designed and the technology transferred to the operational community.

REFERENCES

[1] Moore, M., & Fox, J. (1993). The Virtual Mission Operations Center. In *SOAR '93*. Houston, TX: NASA.